Inverse Reinforcement Learning with Hybrid Weight Tuning and Trust Region Optimization for Autonomous Maneuvering

Yu Shen, Weizi Li, and Ming C. Lin

https://gamma.umd.edu/researchdirections/autonomousdriving/eirl/

Contents

Motivation and Background

Autonomous driving

- Enable vehicles drive safely to the goal with minimal or no human control
- Improve safety and efficiency

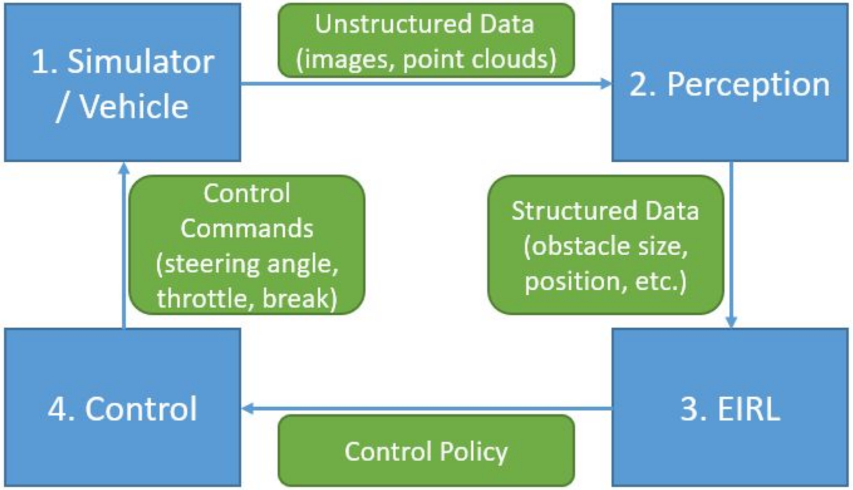
Our Contributions

- **An Enhanced IRL Algorithm (IRL-HWT)**
  - ❖ utilize non-uniform prior with trust region optimization
  - ❖ reuse the model parameters for continuous training,
  - ❖ adopt the “learning from accidents” using expert demonstration and simulation data

- **A Novel Autonomous Driving Pipeline**
  - ❖ context-aware multi-sensor perception
  - ❖ enhanced inverse reinforcement learning

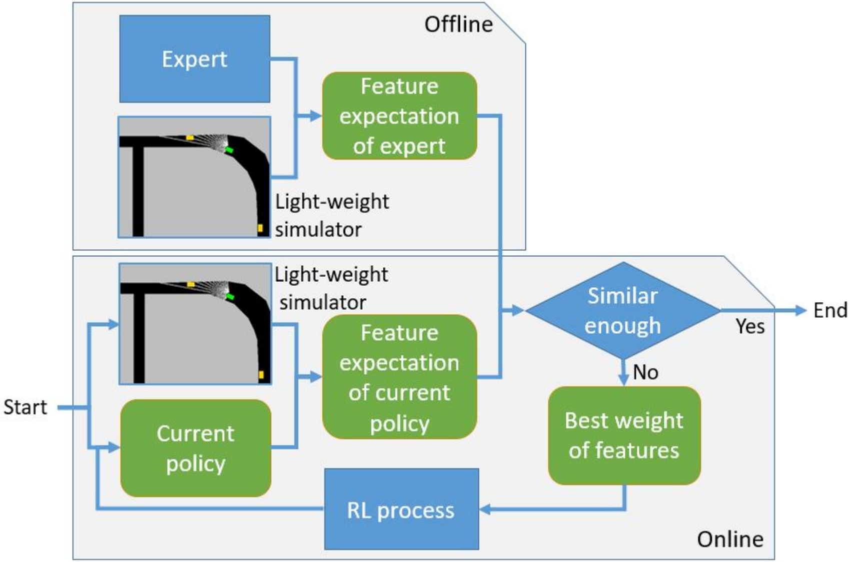
Overall Pipeline

- At each step, the simulator generates unstructured data (images, point clouds)
- These data are processed by Perception Module to produce structured data
- IRL-HWT module then use data to learn a control policy to drive the vehicle



IRL-HWT Pipeline

- *Offline*: Collect experts' trajectories, compute the feature expectation
- *Online*: Compute the feature expectation of the current policy. If similar with the expert policy feature expectation, then stop. Else calculate a new reward and learn a new policy, iteratively.



IRL-HWT Algorithm

- Non-uniform prior
- Reused model parameters
- Learning from accidents

Variable	Description
π	Policy
θ	Model parameters
μ	Feature expectation
w_m	Manually set weights
w_l	Weights to be learnt
$\phi(s)$	Features of a state

Algorithm 1: Inverse Reinforcement Learning with Hybrid Weight Tuning (IRL-HWT)

Result: policy $\pi^{(i)}$
Initialization: Calculate $\mu(\pi_E)$ with expert trajectories;
Set $i = 0$, set $\epsilon, \gamma, \alpha, b_u, b_l, c_u, c_l, p$;
Randomly set the model parameters $\theta^{(0)}$ for $\pi^{(0)}$;
Compute $\mu(\pi^{(0)})$;
Set $w_m^{(0)}$ such that $\|w_m^{(0)}\|_2 < 1$ (initial reward weights), $w_l^{(0)} = \mathbf{0}$;
Compute $\epsilon^{(0)} = (w^{(0)})^T(\mu(\pi_E) - \mu(\pi^{(0)}))$, where $w^{(0)} = [w_m^{(0)} w_l^{(0)}]$;
Set $\Delta^{(0)}$;
while $\epsilon^{(i)} > \epsilon$ **do**
    Set $i = i + 1$;
    Compute the reward function $R = ((w^{(i-1)})^T\phi)$;
    Using R and $\theta^{(i-1)}$ in RL to compute an optimal policy $\pi^{(i)}$;
    Compute $\mu(\pi^{(i)})$;
    Solve Optimization 1 with $\Delta = \Delta^{(i-1)}$, and get solution $\epsilon^{(i)}$ at $w^{(i)}$;
    if Eq. 2 is True **then**
        Accept;
    else
        Reject, solve Eq. 1 with $\Delta = 0$, and update $\epsilon^{(i)}$ and $w^{(i)}$;
    end
    Set $\Delta^{(i)}$ with Eq. 3;
end

Test Scenes

- Scene 1: Open space with only moving vehicles;
- Scene 2: City street with only static obstacles;
- Scene 3: City street with static obstacles and moving vehicles.

Result Comparison

- l_{final} shows safe trajectory length (in meters)
- s_{final} shows how many checkpoints the AV can achieve (number * 100)
- Our method achieves the **highest scores** and enables the AV to *drive safely* **10x further** than the other methods

Method	$l_{final,1}$	$s_{final,2}$	$l_{final,2}$	$s_{final,3}$	$l_{final,3}$
IM	105.6	77.4	53.7	60.1	44.7
IRL	228.8	110.7	69.4	59.7	33.2
GAIL	49.1	103.0	69.9	52.8	35.1
AIRL	74.1	119.9	73.6	83.6	50.7
Ours	**276.3**	**205.8**	**748.4**	**177.3**	**324.2**

Result Comparison

- l_{final} -- safe trajectory length (in meters)
- s_{final} -- number of checkpoints the AV can achieve (number * 100)
- Our method can utilize reward functions (domain knowledge) and expert data to achieve higher performance

Method	$l_{final,1}$	$s_{final,2}$	$l_{final,2}$	$s_{final,3}$	$l_{final,3}$
RL (reward)	72.9	99.8	59.7	59.1	39.7
IRL (expert)	228.8	110.7	69.4	59.7	33.2
Ours (expert+reward)	**276.3**	**205.8**	**748.4**	**177.3**	**324.2**

Feature Effectiveness

- *Non-uniform prior* can reduce the number of collision.

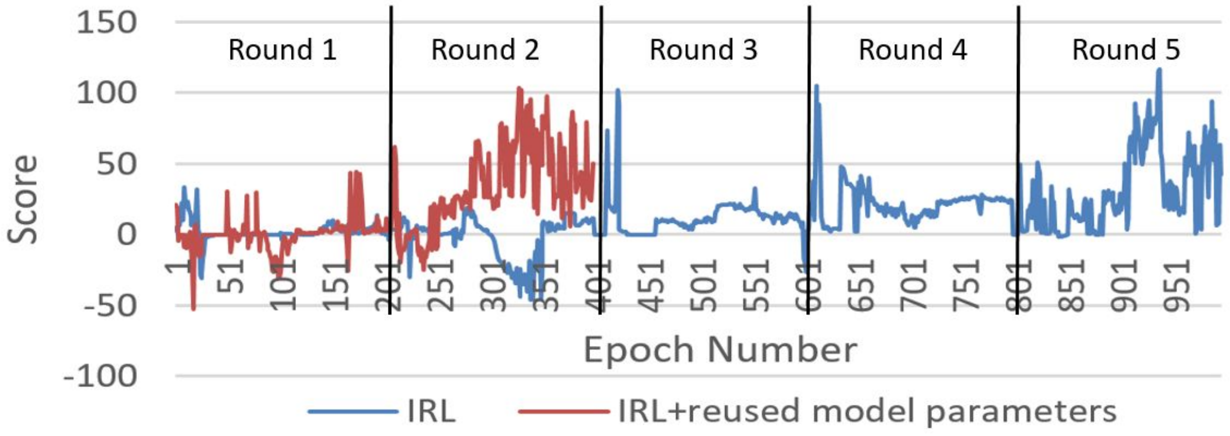
Number of collisions in three different scenes within 10,000 steps: IRL vs. EIRL (IRL+non-uniform prior). Our EIRL can reduce the number of collisions ***up to 41%***.

Model	Scene 1	Scene 2	Scene 3
IRL	35	58	111
IRL+non-uniform prior	**33**	**41**	**93**

Feature Effectiveness

- *"Reused model parameters"* can improve the training efficiency.

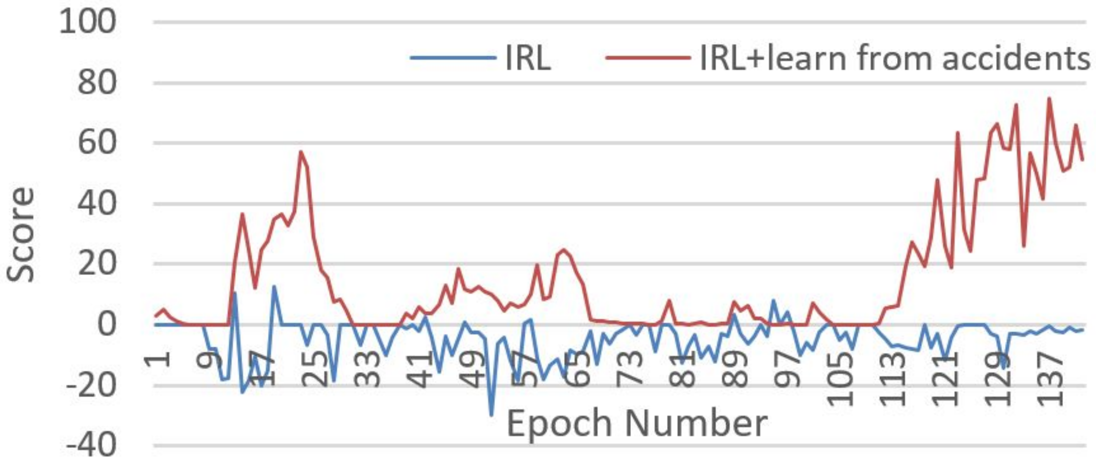
By adding this feature, we can achieve the same score after 2 rounds of training -- otherwise taking 5 rounds of training -- resulting in ***<u>2.5x</u> speedup***



Feature Effectiveness

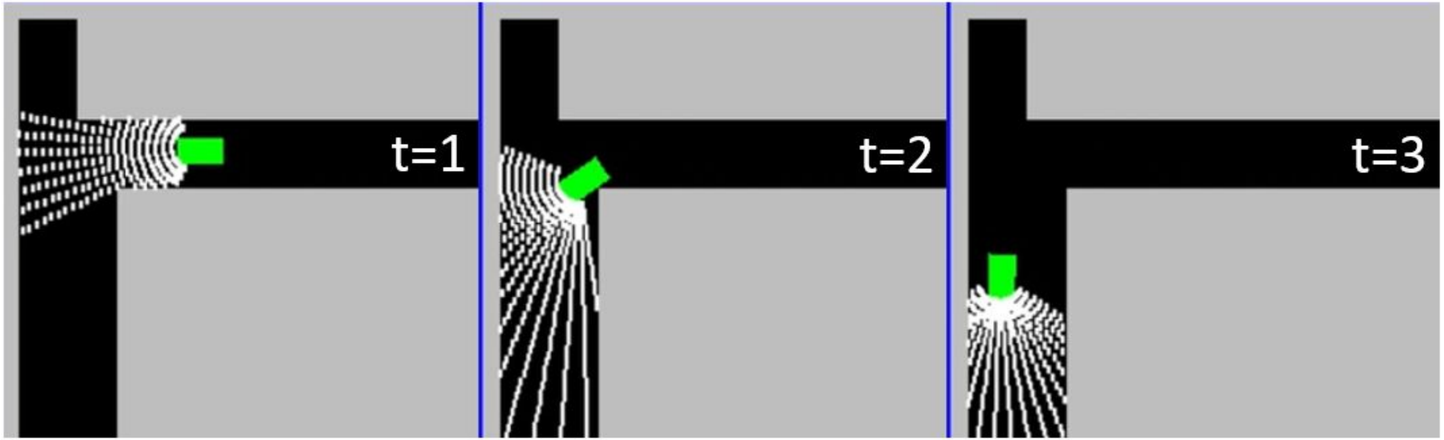
- *"Learning from accidents"* can also improve the training efficiency.

With additional training data, the learning algorithm achieves higher scores up to ***two orders of magnitude*** in near collision scenarios under the same number of epochs.



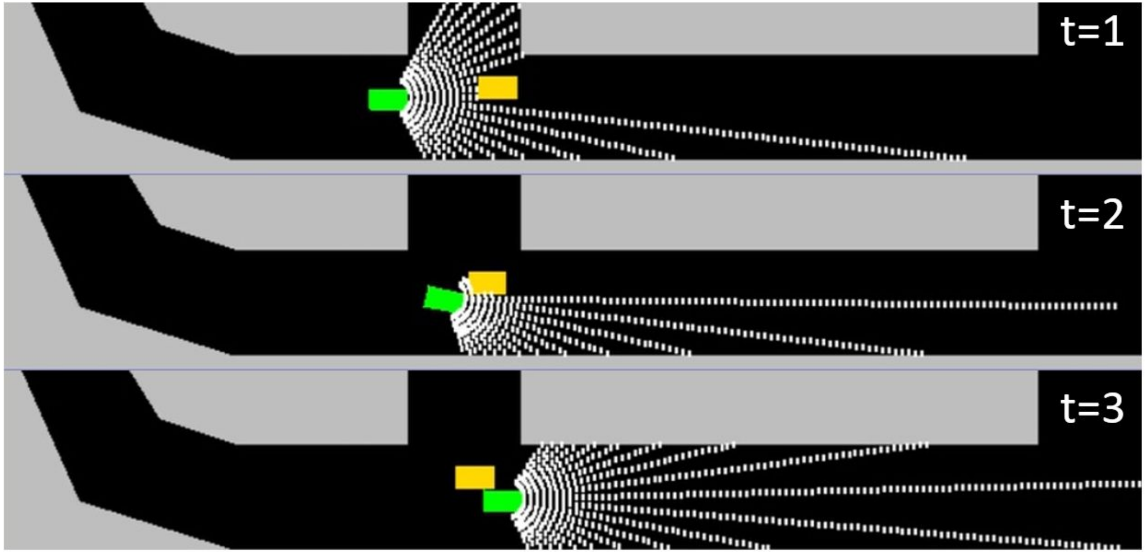
Case Analysis

- ***Static obstacle avoidance***: Our method can enable the car to make a left turn for collision avoidance and resume safe driving.



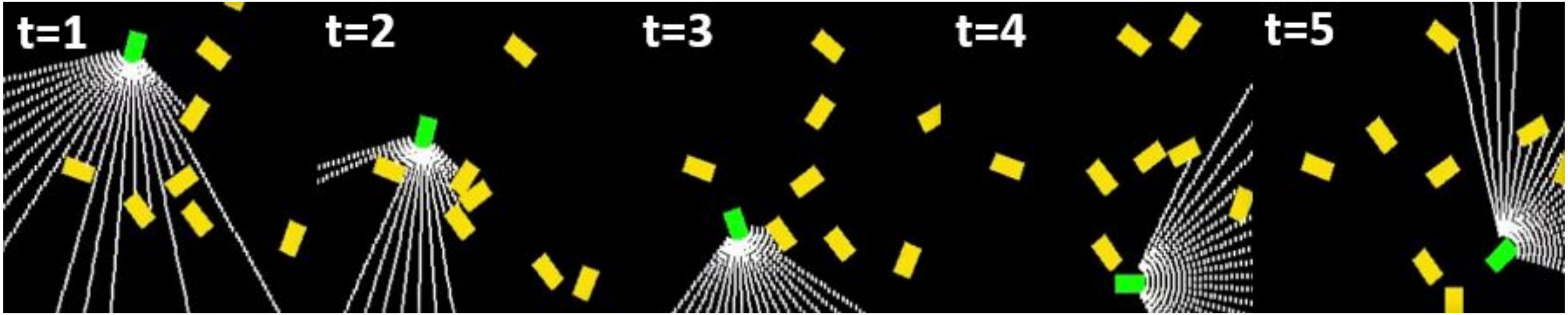
Case Analysis

- ***Static and dynamic obstacle avoidance***: Our car (in green) can avoid another car (in yellow) coming from the opposite direction, while steering away from the static obstacles.

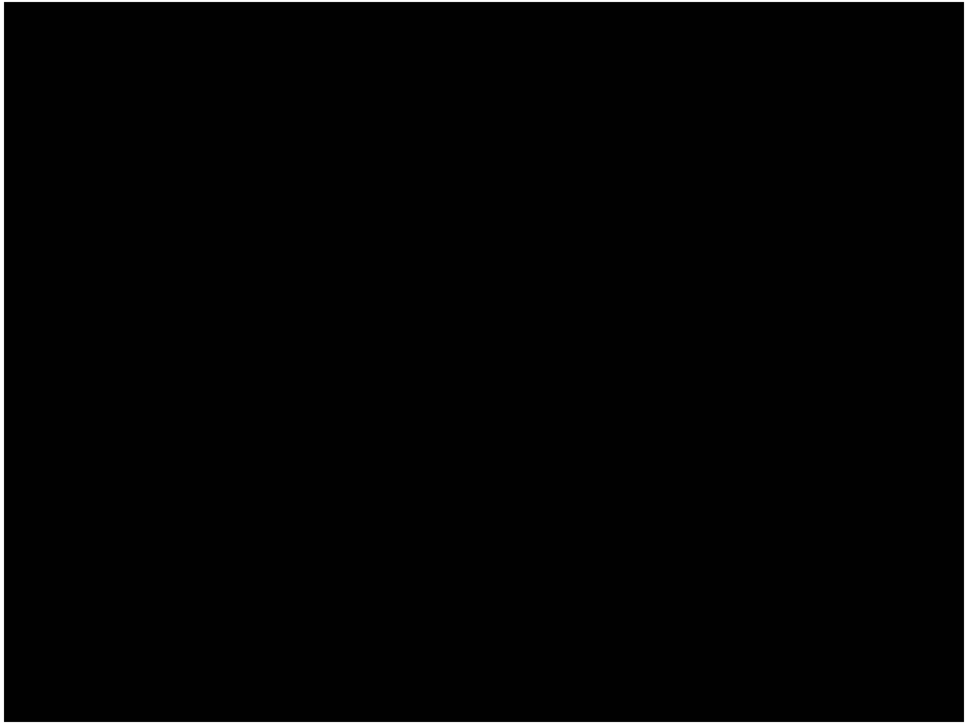


Case Analysis

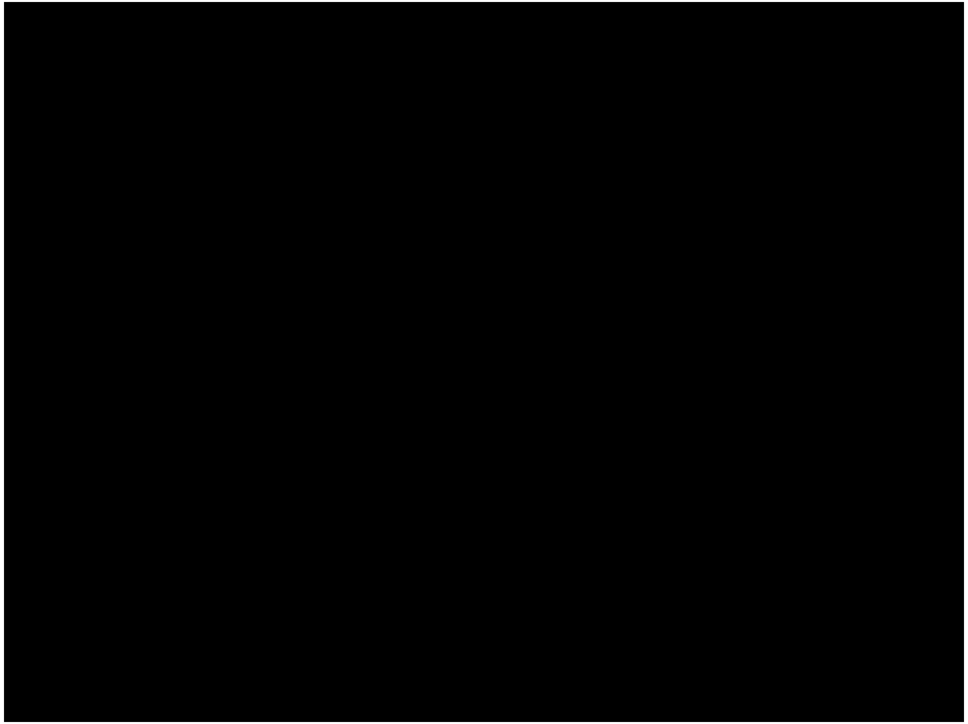
- ***Collision avoidance with multiple dynamic obstacles***: Our method can direct the car (in green) to avoid all nearby cars (in yellow), even when a narrow passage is present



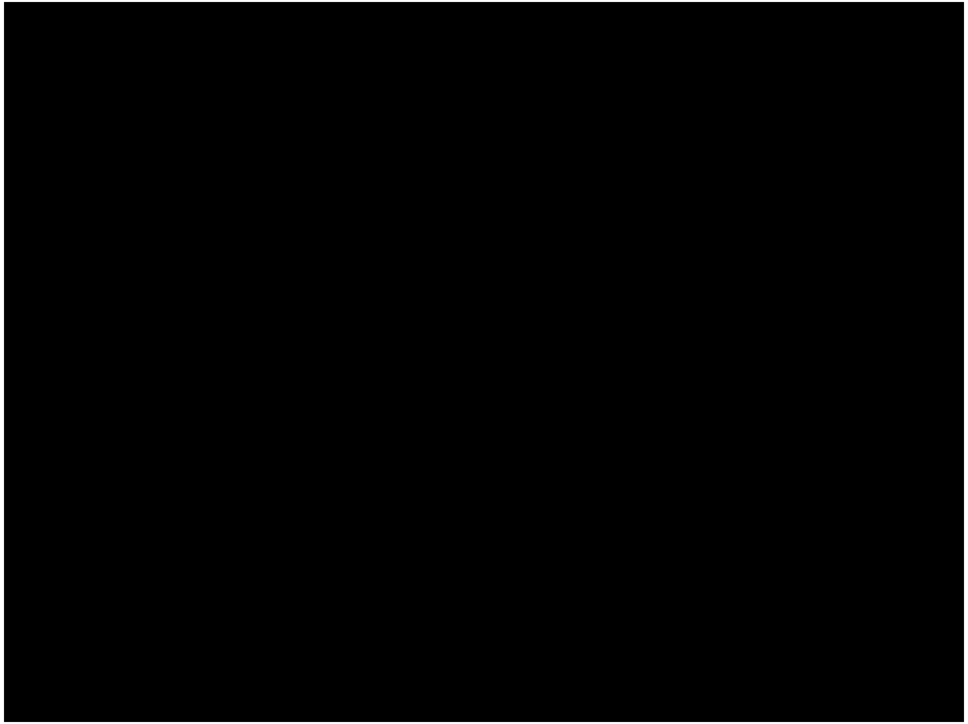
Video Demo: Collision Avoidance in Open Space



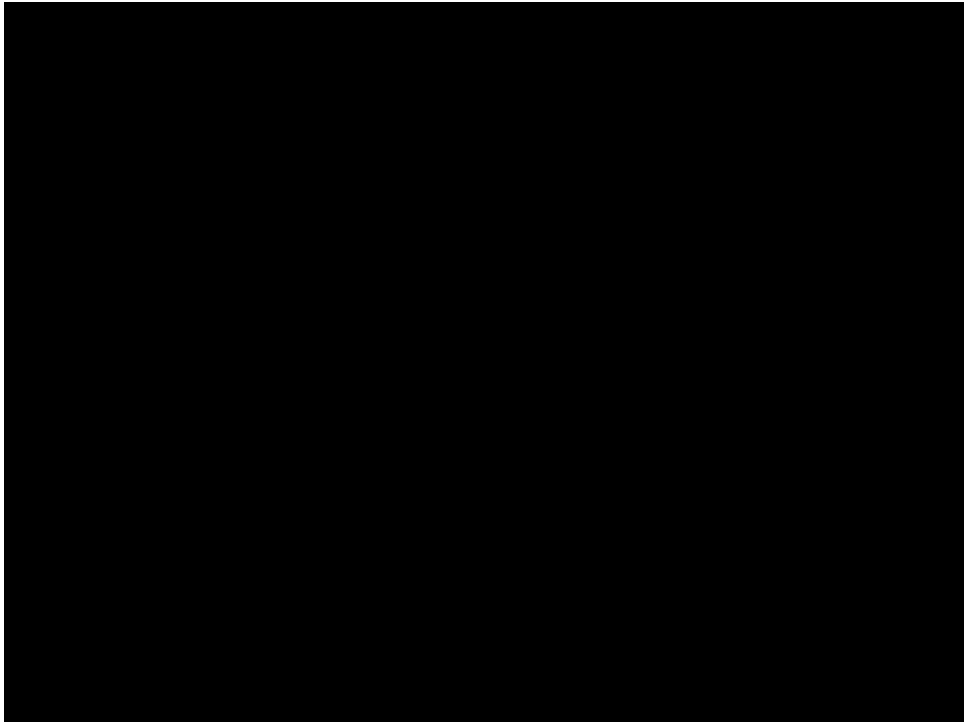
Video Demo: Collision Avoidance on City Streets



Video Demo: Collision Avoidance on City Streets with Other Cars



Video Demo: Demo in Unity



Summary

- **A enhanced IRL algorithm (IRL-HWT)** that can
  - utilize non-uniform prior with trust region optimization
  - reuse the model parameters for continuous training
  - adopt “learning from accidents” by using expert demonstration and simulation data

- Our method can utilize both reward functions (domain knowledge) and expert data to drive safely ***10x longer*** than other methods

Thank you!